CLAIMS

We claim:

Claim 1 (original): An article of manufacture for use in a computer system for translating a path expression in an object oriented query to a relational database outer join, said path expression comprising a navigation path through a relationship in a schema, said article of manufacture comprising a computer-useable storage medium having a computer program embodied in said medium which causes the computer system to execute the method steps comprising:

analyzing each path expression defined in each level of the object oriented query;

identifying each path expression which can be a candidate for a translation to an outer join;

ordering the path expression starting with path expression defined in a FROM clause, adding to the FROM clause path expression, each path expression identified as a candidate for a translation to an outer join, and making the ordered path expressions as input to a select operator for each level of the object oriented query;

grouping the ordered path expressions sequentially based upon on a source-target dependency between ordered path expressions and based upon the identifications as a candidate for a translation to an outer join;

creating a quantifier for each path expression, said quantifier comprising a variable representing a table in a relational database;

replacing each grouped path expression with a corresponding quantifier and related table in a relational database; and

completing a translation of the object oriented query to a relational query.

Claim 2 (original): The article of manufacture of claim 1 wherein the embodied computer program embodied in said medium can further cause the computer system to execute the method steps comprising:

performing optimization on the grouped quantifiers, said optimization identifying quantifiers which can be a candidate for a translation to an inner join;

generating an outer join for each quantifier which remains after optimization a candidate for a translation to an outer join; and

generating an inner join for each quantifier which remains after optimization a candidate for a translation to an inner join .

Claim 3 (original): The article of manufacture of claim 2 wherein the optimization identifies a quantifier as a candidate for a translation to an inner join if a corresponding path expression is used in a FROM clause.

Claim 4 (original): The article of manufacture of claim 2 wherein the optimization identifies a quantifier as a candidate for a translation to an inner join if a LIKE, IN, or BETWEEN operator exists in a WHERE clause containing a corresponding path expression.

Claim 5 (original): The article of manufacture of claim 2 wherein the optimization identifies a quantifier as a candidate for a translation to an inner join if an EQUAL, LESS THAN, GREATER THAN, LESS THAN OR EQUAL, GREATER THAN OR EQUAL, NOT EQUAL, or NOT NULL operator exits in a WHERE clause.

Claim 6 (original): A method of translating a path expression in an object oriented query to a relational database outer join, said path expression comprising a navigation path through a relationship in a schema, said method comprising the steps of:

analyzing each path expression defined in each level of the object oriented query;

identifying each path expression which can be a candidate for a translation to an outer join;

ordering the path expressions starting with path expressions defined in a FROM clause, adding to the FROM clause path expressions, each path expression identified as a candidate for a translation to an outer join, and making the ordered path expressions as input to a select operator for each level of the object oriented query;

grouping the ordered path expressions sequentially based upon on a source-target dependency between ordered path expressions and based upon the identifications as a candidate for a translation to an outer join;

creating a quantifier for each path expression, said quantifier comprising a variable representing a table in a relational database;

replacing each grouped path expression with a corresponding quantifier and related table in a relational database; and

completing a translation of the object oriented query to a relational query.

Claim 7 (original): The method of claim 6 further comprising the steps of:

performing optimization on the grouped quantifiers, said optimization identifying quantifiers which can be a candidate for a translation to an inner join;

generating an outer join for each quantifier which remains after optimization a candidate for a translation to an outer join; and

generating an inner join for each quantifier which remains after optimization a candidate for a translation to an inner join.

Claim 8 (original): The method of claim 7 wherein the optimization identifies a quantifier as a candidate for a translation to an inner join if a corresponding path expression is used in a FROM clause.

Claim 9 (original): The method of claim 7 wherein the optimization identifies a quantifier as a candidate for a translation to an inner join if a LIKE, IN, or BETWEEN operator exists in a WHERE clause containing a corresponding path expression.

Claim 10 (original): The method of claim 7 wherein the optimization identifies a quantifier as a candidate for a translation to an inner join if an EQUAL, LESS THAN, GREATER THAN, LESS THAN OR EQUAL, GREATER THAN OR EQUAL, NOT EQUAL, or NOT NULL operator exits in a WHERE clause.

Claim 11 (original): A computer system for translating a path expression in an object oriented query to a relational database outer join, said path expression comprising a navigation path through a relationship in a schema, said computer system comprising:

computer program instructions for analyzing each path expression defined in each level of the object oriented query;

computer program instructions for identifying each path expression which can be a candidate for a translation to an outer join;

computer program instructions for ordering the path expressions starting with path expressions defined in a FROM clause, adding to the FROM clause path expressions, each path expression identified as a candidate for a translation to an outer join, and making the ordered path expressions as input to a select operator for each level of the object oriented query;

computer program instructions for grouping the ordered path expressions sequentially based upon on a source-target dependency between ordered path expressions and based upon the identifications as a candidate for a translation to an outer join;

computer program instructions for creating a quantifier for each path expression, said quantifier comprising a variable representing a table in a relational database;

computer program instructions for replacing each grouped path expression with a corresponding quantifier and related table in a relational database; and

computer program instructions for completing a translation of the object oriented query to a relational query.

Claim 12 (original): The computer system of claim 11 further comprising:

computer program instructions for performing optimization on the grouped quantifiers, said optimization identifying quantifiers which can be a candidate for a translation to an inner join;

computer program instructions for generating an outer join for each quantifier which remains after optimization a candidate for a translation to an outer join; and

computer program instructions for generating an inner join for each quantifier which remains after optimization a candidate for a translation to an inner join.

Claim 13 (original): The computer system of claim 12 wherein the optimization identifies a quantifier as a candidate for a translation to an inner join if a corresponding path expression is used in a FROM clause.

Claim 14 (original): The computer system of claim 12 wherein the optimization identifies a quantifier as a candidate for a translation to an inner join if a LIKE, IN, or BETWEEN operator exists in a WHERE clause containing a corresponding path expression.

Claim 15 (original): The computer system of claim 12 wherein the optimization identifies a quantifier as a candidate for a translation to an inner join if an EQUAL, LESS THAN, GREATER THAN, LESS THAN OR EQUAL, GREATER THAN OR EQUAL, NOT EQUAL, or NOT NULL operator exits in a WHERE clause.